МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

# **ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ПРОЕКТНОЙ РАБОТЕ**

# **«GCodeGeneratore: оконное приложение для формирования управляющих программ фрезерных станков ЧПУ»**

# **ПО ДИСЦИПЛИНЕ**

# **“ПРОЕКТНО- ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА”**

09.03.02 Информационные системы и технологии профиль «Разработка IT-продуктов и информационных систем»

квалификация бакалавр, форма обучения очная

Подготовил: Леус Александр Сергеевич (Команда №10)

1. **Цели и задачи проекта.**

Взаимодействие с современными ЧПУ-станками реализовано с помощью управляющих программ - файлов с G-кодом (далее УП). Существующее ПО для формирования УП обладает недостатками: большая стоимость приобретения, сложность в обучении, большие трудозатраты в работе.

Например, фрезерование резьбы при использовании профессиональных CAD-CAM систем состоит из следующих этапов:

* Построение 3D моделей для детали, заготовки, фрезы и оснастки станка.
* Настройка параметров станка: выбор системы координат и нуля на модели, установка значения безопасной высоты, расстояний отвода и др.
* Выбор и настройка режима фрезерования: более 30 параметров для каждой операции.
* Расчет траекторий и обработка их постпроцессором(свой для каждого станка).

Т.О. подготовка G-кода даже для простейших операций может занимать более двух часов, а постпроцессор для необходимой модели станка может отсутствовать.

**Цель проекта** – сокращение финансовых затрат, оптимизация работы по формированию УП для 3-х осевых ЧПУ станков, обеспечение совместимости с хоббийной техникой (ЧПУ на логике GRBL и аналогов).

**Задачи проекта** – разработка ПО для исключения CAD-CAM систем при формировании УП для машин ЧПУ (для личного использования).

1. **Этапы реализации проекта.**

* **Составление ТЗ:** формулирование существующих проблем, определение общей концепции и предпочтительного функционала.
* **Поиск готовых решений:** анализ доступного ПО и его тестирование. Требуемый результат получен не был, принято решение о разработке собственного приложения.
* **Изучение ЧПУ фрезерования:** изучение информации о методах управления ЧПУ станками, анализ работы CAM-систем и постпроцессоров, изучение базовых команд G-кода, анализ совместимости с микроконтроллерами станков, определение наиболее часто используемых видов фрезерования и сбор данных по основным режимам обработки.
* **Определение функциональности:** исходя из личных возможностей и умений был выбран формат оконного приложения. Определены основные виды фрезерования: плоскость, отверстие, резьба. Составлен перечень необходимых настроек и вводных данных.
* **Разработка логики траекторий:** для каждого вида обработки разработан алгоритм для определения координат шпинделя станка для каждой точки траектории, с допуском в 0.01мм и 1 градус (с учетом диаметра фрезы, попутного/встречного направления фрезерования, направления для резьбы и др.)
* **Разработка приложения:** разработан оконный интерфейс приложения на C# Forms, сформирован программный код для логики траекторий.
* **Анализ работы:** сформированные приложением G-коды загружены в имеющийся ЧПУ-станок, подтверждена корректность работы и совпадение геометрических размеров готовой детали заданным параметрам при составлении УП.
* **Подготовка презентации.**
* **Использование приложение на практике:** приложение успешно используется при выполнении единичных работ в домашней мастерской (изготовление станочной оснастки, корпусов подшипников; подготовка базы/центров деталей; нарезка нестандартной резьбы и др.)

1. **Стек технологий.**

Для разработки приложения был использован язык программирования C#.

Выбор обусловлен отсутствием необходимого опыта работы с другими инструментами, а также отсутствием команды, что исключает делегирование проблемных вопросов. При этом C# подходит для решения всех задач проекта.

**Общий перечень используемых технологий:**

* **С# (Forms) –** разработка оконного приложения.
* **Candle –** удобная программа для передачи G-кода контроллеру станка.
* **GRBL, Arduino и аналоги –** управление железом имеющегося в наличии станка.
* **Solidworks, SolidCam и аналоги –** профессиональные CAD-CAM системы.

1. **Описание проблем, путей их решения.**

* **Основной проблемой** при реализации проекта считаю недостаток знаний и умений, полученных на 1-м семестре обучения. К концу второго семестра освоен базовый функционал C#, получены навыки создания оконных приложений, что и стало решением вопроса.
* **Недоступность документации в сфере металлообработки.** Для большинстваПО и станков документация недоступна для свободного изучения. Решением стал поиск информации на специализированных форумах.
* **Отсутствие опыта разработки приложений**. Опыт разработки приложений был приобретен в процессе разработки приложения.

1. **Описание функциональности проекта.**

Разработанное приложение позволяет пользователю быстро и удобно генерировать УП для базовых видов фрезерной обработки.

Интерфейс программы – оконное приложение. Для выбора геометрии используется три кнопки. Каждая кнопка открывает окно, где необходимо ввести параметры фрезерования. При нажатии на кнопку «Сгенерировать G-код» приложение создает файл с управляющей программой. Реализована проверка корректности введенных пользователем значений. На основе полученных данных приложение рассчитывает координаты каждой точки траектории с допуском в 0.01мм, формирует команды G-кода, и записывает их в файл.

* **Обработка плоскости (торцевое фрезерование).**

Пользователь задает параметры: длину, ширину, общую глубину участка обработки, шаг по Z, перекрытие инструмента, диаметр фрезы, а также скорости подач по всем осям.

* **Обработка отверстия (круглый карман).**

Доступные параметры: координаты центра окружности, размеры отверстия, шаг по Z, перекрытие инструмента, диаметр фрезы, скорости подач.

Отдельно добавлена настройка углового заглубления фрезы для снижения нагрузок на станок.

* **Обработка резьбы.**

Доступные параметры: координаты центра и диаметр резьбы, шаг резьбы, диаметр фрезы, скорости подач.

Отдельно добавлены: настройка многозаходности резьбы, выбор направления (левая/правая резьба), выбор режима обработки (встречное/попутное фрезерование).

1. **Дальнейшая судьба проекта.**

Основная концепция приложения, исключающая 3D моделирование, останется без изменений. Доработки будут реализованы по мере необходимости в работе мастерской. В настоящее время сформулированы основные задачи.

**Планы модернизации интерфейса:**

* Графическое отображение траекторий с анимацией движения фрезы;
* Возможность сохранения наборов параметров;
* Расчет и отображение справочной информации о режимах резания (окружная скорость фрезы, съем на зуб, усилие резания и др.);
* Возможность выбора директории и названия файла при генерации УП;
* Всплывающие окна со справкой и рекомендациями по доступным параметрам.

**Планы модернизации функционала:**

* Обработка пользовательских ошибок;
* Увеличение количества настроек траекторий (растровая, по спирали, выбор точки начала фрезерования, плоскости под углом/вертикальные и т.д.);
* Возможность добавления элементов в очередь и формирование общей УП;
* Новые элементы: внешняя резьба, стандартные прямозубые шестерни, шестиугольники и др.

**Новый проект:** система управления для полуавтоматической установки для биопсии (многозвеньевая робот рука) – на основе наработок представленного проекта.

1. **Техническое описание проекта.**

Проект реализован на C# Forms и состоит из 4 форм. Масштабирование запрещено.

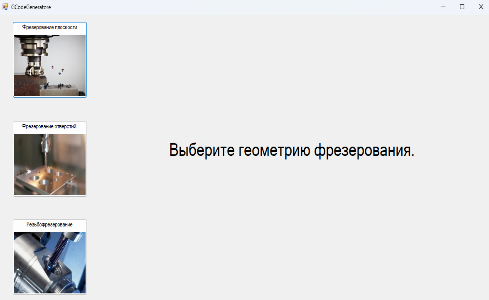
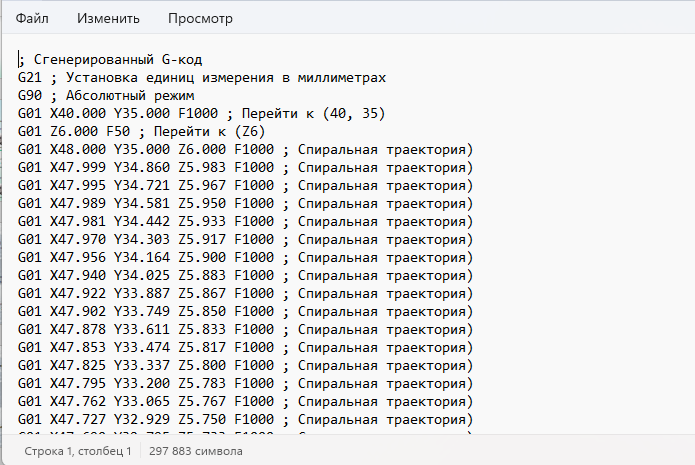
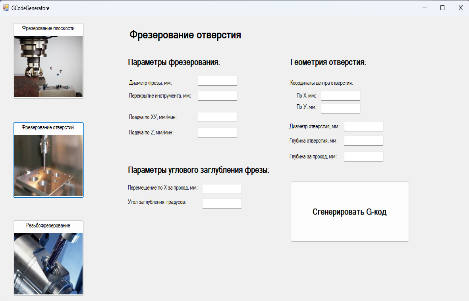
Первая форма – стартовое окно. Три кнопки отвечают за вызов соответствующей ей формы. Формы 2, 3, 4: отвечают за выбор типа фрезерования, жестко закреплены внутри формы 1.

В каждой форме реализован свой алгоритм расчета и преобразования в G-код координат точек траекторий (1 метод в каждой форме, учитывающий все параметры).

В формах 2,3,4, пользователю нужно заполнить параметры и нажать на кнопку генерации УП.

Запуск приложения реализован через ярлык «GCodeGeneratore» (расположен в директивной папке проекта).

Схема работы приложения:



Запуск приложения.

Выбор нужного режима.

Ввод параметров. Отработка ошибок (при возникновении).

Клик на кнопку «Сгенерировать G-код».

Готовый файл с УП.

Расположение: GCodeGeneratoe/bin/debug

1. **Инструкция для пользователя.**
2. Ссылка на репозиторий GitHub: [**https://github.com/LeusAlex/PIR-GCodeGenerator**](https://github.com/LeusAlex/PIR-GCodeGenerator)
3. Запустить приложение через ярлык GCodeGeneratore
4. Выбрать необходимую геометрию обработки.
5. Ввести или выбрать значения для всех доступных параметров. (Рекомендованы целые положительные числа).
6. Самостоятельно проверить корректность введенных значений (диаметр фрезы меньше диаметра отверстия, шаг по Z меньше общей глубины и т.п.).
7. Нажать кнопку «Сгенерировать G-код».
8. Файл с УП появится в папке «GCodeGenerator/bin/debug». Для разных типов обработки создаются файлы с разным названием. При повторном создании УП одного типа файл перезапишется.
9. УП готова. Можно загружать файл в станок/транслирующую программу.
10. Для визуального контроля сгенерированной траектории прикладываю в репозиторий программу Candle. Запустите через ярлык, перетащите в окно Candle сформированный файл УП, на главном экране появится визуальное представление траектории.
11. Контакты техподдержки: telegram [@leus\_a](https://t.me/leus_a)